

GA 와 GP 를 이용한 모듈라 로봇 이동 제어

Locomotion Control of Modular Robot Using GA and GP

장재영 · 현수환 · 서기성

Jaeyoung Jang and Soohwan Hyun and Kisung Seo

서경대학교 전자공학과

E-mail: geruld@naver.com, xjavalov@shhyun.com, ksseo@skuniv.ac.kr

요 약

모듈라 뱀형 로봇은 고장에 대한 강인성과 환경에 유연한 이동 특성을 가지고 있으나, 제어가 어렵다는 단점이 있다. 진화연산을 로봇에 이용한 많은 연구가 진행되어 왔지만, 어떤 기법의 진화연산이 문제에 더 적합하고, 높은 성능을 얻을 수 있는지에 대한 비교는 거의 이루어지지 않고 있다. 본 논문은 두 가지 대표적인 진화기법인 GA와 GP를 이용하여 모듈라 뱀형 로봇의 이동 제어를 수행하였다. 대상 로봇은 H/W로 구현이 가능한 실제 모듈로 구성되었고, Webots를 사용하여 시뮬레이션 실험을 수행하였으며, GA와 GP 기법에 의한 결과를 비교 분석하였다.

Key Words : Modular Robot, Locomotion, Genetic Algorithm, Genetic Programming

1. 서 론

모듈라 로봇은 독립적인 단위 구성 요소인 모듈을 원하는 기능을 수행할 수 있도록 특정한 형태로 연결하여 만든다[1,2].

구성 형태에 따라 뱀형, 거미형, 루프형, 보행형 등 다양한 종류의 로봇 구현이 가능하다. 또한, 모듈의 특성상 쉽게 연결 구성을 바꿀 수 있으므로 재구성(reconfiguration)이 가능하며, 일부 모듈의 고장이나 결함에도 불구하고 상당한 강인성을 가지는 장점이 있다.[1,2]

모듈라 뱀형 로봇은 다수의 모듈을 일렬로 연결하여 구성된 이동로봇으로서, 기존의 바퀴나 보행 로봇에 비하여 다음과 같은 장점을 가지고 있다. 일부분이 고장 났을 경우에 정상적인 모듈들만으로도 동작이 가능하며, 환경이 불안정 하거나, 장애물이 있는 지형에서도 유연하게 움직일 수 있는 특성을 갖는다. 그러나 기존의 바퀴나 보행 로봇에 비하여 이동속도가 느리고 제어가 어렵다는 단점이 있다.[3,4]

기존의 모듈라 뱀형 로봇의 이동제어에 사용된 방식으로는 삼각함수를 이용해 로봇 전체의 움직임을 표현한 후, 그것을 각각의 모듈에 대해 일정 주기만큼 수평이동하여 지속적으로 신호를 입력하는 방식이 있다.[5] 그리고 CPG(central pattern generator)와 간단한 최적화 연산을 이용해 기어가는 동작과 헤엄치는 동작을 생성한 방법 등이 있다.[3]

그러나 사람이 직접 수학적으로 이동방법(locomotion)을 설계하는 방식은 예측 불가능한 환경 조건이나 복잡한 로봇의 형태, 능력이 저하된 모듈에서의 적응 등의 상황에 대해서는 설계가 어렵다. 이에 비해, 진화 연산을 이용하면 사람에 의한 설계에서는 예측할 수 없는 보다 다양한 이동(locomotion) 방식의 생성이 가능하다.

이러한 진화연산 기법을 사용한 모듈라 로봇 연구가 수행되어 왔고, 시뮬레이션이나 실제 구현 면에서 괄목할

만한 성과를 얻고 있다.[1-4]

그러나, 모듈라 로봇 이동에 어떤 기법의 진화연산이 더 적합하고, 높은 성능을 얻을 수 있는지에 대한 비교는 거의 이루어지지 않고 있다.

따라서, 본 논문에서는 진화 연산중 대표적인 두 기법인 GA[6]와 GP[7]를 사용하여 뱀형 모듈라 로봇의 이동 제어를 설계하고 성능을 비교하였다. 대상 로봇은 H/W로 구현이 가능한 실제 모듈로 구성되었으며, 물리적 특성의 시뮬레이션 가능한 Webots[8]을 사용하여 시뮬레이션 실험을 수행하였다.

2. 모듈라 뱀형 로봇의 이동제어

2.1 모듈라 뱀형 로봇 모델

본 논문의 실험에서 사용된 모듈은 BIRG (Biologically Inspired Robotics Group)의 모듈라 로봇의 자가 가구 구성(Modular robotics for adaptive and self-organizing furniture)에서 사용되는 기본 모듈로써 명칭은 roombot이다[9].

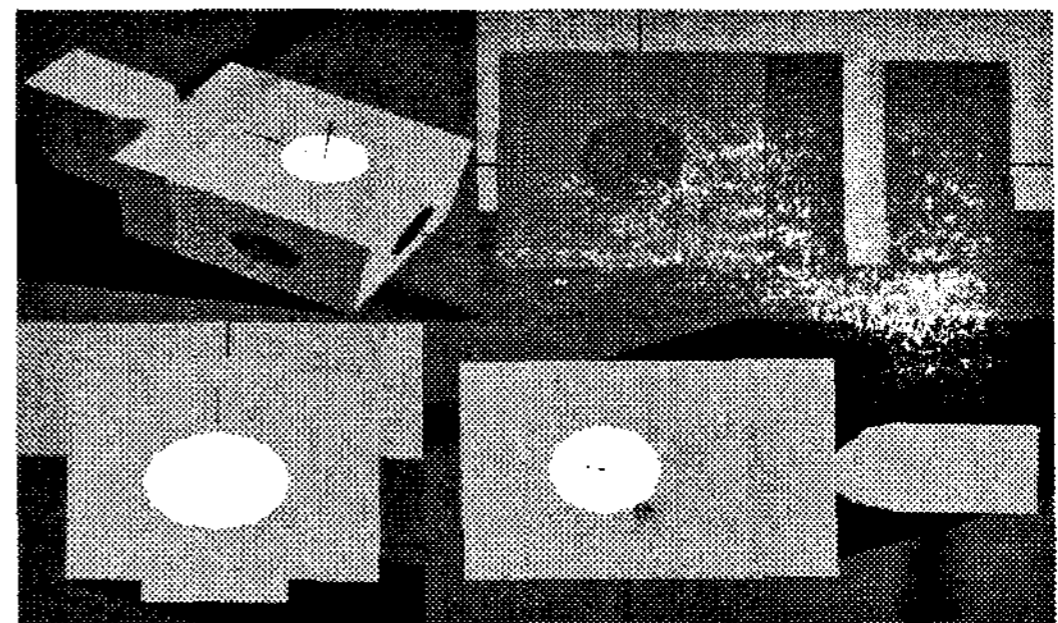


그림 1. Roombot의 모습 (a) 전체적인 모습, (b) 측면에서 본 모습, (c) 정면에서 본 모습, (d) 위에서 본 모습.

모듈은 1개의 관절과 컨트롤러와 전원, 6개의 커넥터를 가지며, 1개의 DOF를 가진다. 모든 관절은 $-1.57 \sim 1.57$ 사이의 동일한 각도 제한을 가진다.

뱀형 로봇은 모듈 8개를 사용하여 구성하였으며, 앞 모듈의 후면 커넥터와 뒤 모듈의 전면 커넥터 2개를 차례대로 결합하는 방식으로 각 모듈들을 연결하여 구성하였다.

2.2 모듈라 뱀형 로봇의 이동 방법

모듈라 뱀형 로봇의 이동 방법으로는 크게 수직적인 관절 움직임을 가지는 방법과 수평적인 관절 움직임을 가지는 방법, 수직과 수평 모두 움직일 수 있는 방법이 있다.

수평적인 관절의 움직임을 가지는 방법으로는 몸체의 측면 방향으로 전진하는 사이드와인더 방식[4,10], S모양으로 움직이는 측면 파동 방법(Serpentine)[3,10] 등이 있다.

수직적인 관절의 움직임을 가지는 방법은 일반적인 애벌레의 움직임과 같은 Caterpillar방식[5,11] 꼬리부분으로 몸체를 밀어 이동하는 Inching 방식 등이 있다[10].

그 외에 수평 수직 복합적인 움직임의 방법으로는 Helix Theory방법[12]이나 screw방법[10,11] 등이 있다.

3. GA 와 GP 기반 이동 제어

GA(Genetic Algorithm)[6]와 GP(Genetic Programming)[7]는 진화연산의 한 종류로서, 유전자(Chromosome)를 표현하는 방식에 서로 차이가 있다.

GA는 Bit String이나 실수형태의 염색체를 이용하고, GP는 트리구조의 염색체를 이용한다. 이 차이로 인해 GA는 고정된 크기의 염색체로 구성되며, 주로 수치최적화에 적합하다. 이에 비해 GP는 트리구조를 사용하기 때문에 염색체의 크기가 가변이며, 구조적인 설계문제나 임의의 차수나 조합적으로 구성되는 복잡한 함수를 탐색하는 것에 적합하다.

3.1 GA 기반 이동 제어

GA를 사용한 방법에서는 뱀 로봇을 제어할 수식의 파라미터를 결정한다. 식 (1)과 같이 주어진 사인함수에 대해서, 진폭(A)과 주기(ω), 위상(ϕ)에 변화를 주어 각 모듈에 대한 관절 값을 제어 한다.

$$A \sin(\omega(t) + \phi) \quad (1)$$

$(-1.57 \leq A \leq 1.57 \quad -1.5 \leq \omega \leq 1.5 \quad -6.28 \leq \phi \leq 6.28)$

한 관절(모듈)에 대해서 진폭(A), 주기(ω), 위상(ϕ)의 3개의 필드를 가지며, 개체는 8개의 관절로 이루어지므로 총 24개의 필드로 구성된 염색체를 사용한다.

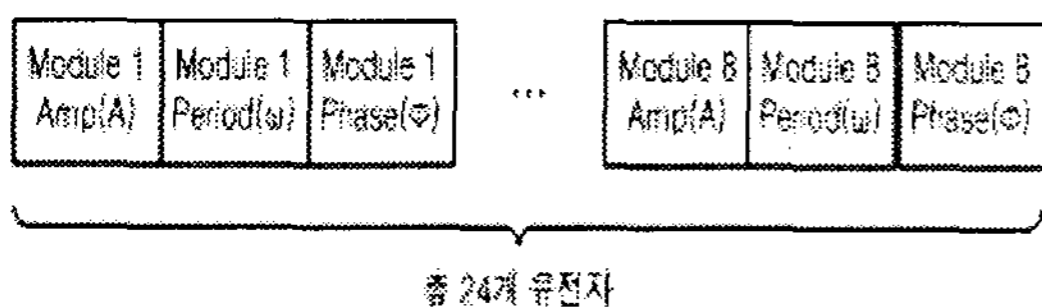


그림 2. 유전자(Chromosome)의 구성.

3.2 GP 기반 이동 제어

GA 기반 방법이 고정된 구조의 사인함수에 파라미터만을 변화시켜 이동 제어식을 탐색하는데 비하여, GP 방식에서는 각 모듈이 움직이는 관절 궤적이 임의의 함수로

구성된다. 즉, 그림 2와 같이 각 관절에 대한 궤적 함수가 GP에 의해서 생성, 진화되는 트리를 통해 탐색된다.

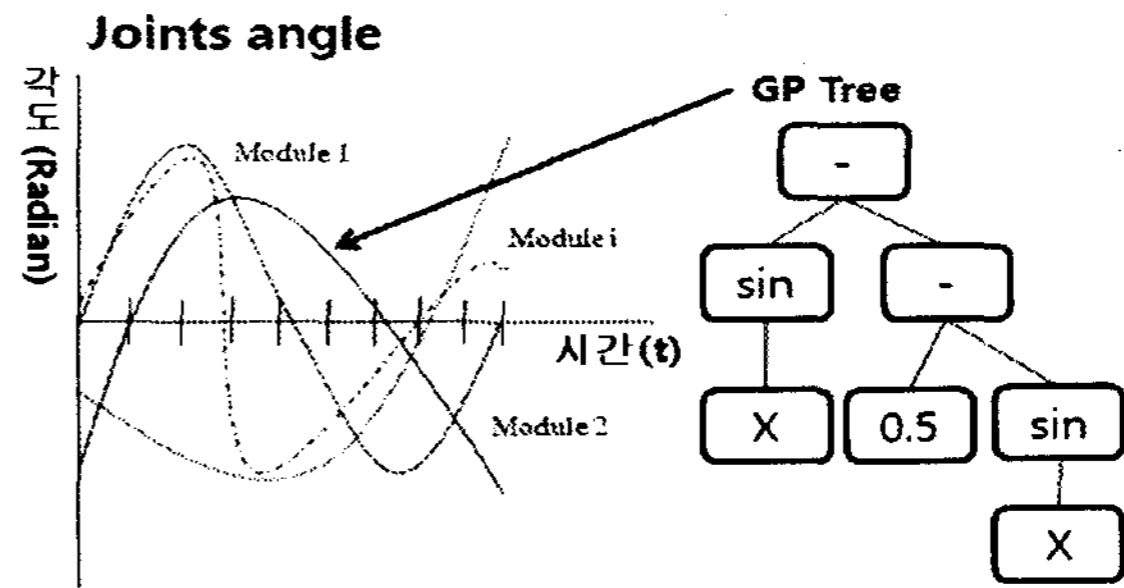


그림 3. GP Tree를 통한 움직임 생성.

각 관절의 독립적인 제어를 위해서, 일반적인 GP의 단일트리 방식이 아닌 멀티트리 방식을 이용함으로써 더 효율적인 연산을 꾀하였다.

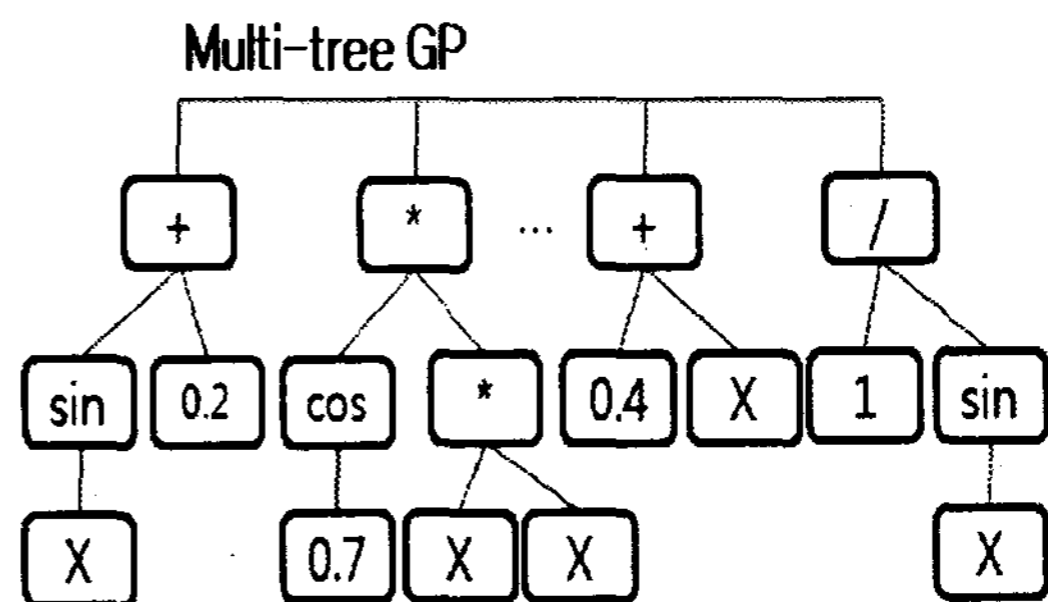


그림 4. 멀티트리 GP 방식.

GP 트리의 구성에 이용된 함수와 터미널은 다음과 같다.

$$F = \{ \sin, \cos, +, -, *, / \}$$

$$T = \{ X, \text{erc} \}$$

4. 컴퓨터 시뮬레이션

4.1 Webots 시뮬레이션 환경

시뮬레이션 환경은 Cyberbotics 사의 Webots[8]을 사용하였다. Webots은 모바일 로봇에 대한 모델링, 프로그래밍, 그리고 시뮬레이션 기능을 제공하는 모바일 로봇 시뮬레이션 s/w 이다. 주요 기능은 다음과 같다. 각종 센서와 액츄에이터에 대한 라이브러리 제공, 정확한 physics 시뮬레이션을 위한 ODE(Open Dynamics Engine) 라이브러리 제공, 그리고 실제 모바일 로봇에 컨트롤러를 트랜스퍼 할 수 있는 기능을 가지고 있다.

4.2 적합도 함수 및 GA, GP 파라미터

적합도 함수는 직진거리(x)와 벗어난 거리(z)의 차이를 이용했다. 본 실험에서는 직진거리에 좀 더 많은 가중치를 두어 적합도 함수를 구성하였다.

$$fitness = (0.9 \times x - 0.4 \times |z|)$$

GA 프로그램으로는 기본적인 SGA[6]가 사용되었다. GP 엔진으로 lil-gp[13]를 사용하였다.

GA와 GP에 공통으로 이용된 파라미터는 다음과 같다.

Number of generations : 50
 Population sizes : 20, 30
 Crossover : 0.9
 Mutation : 0.1
 Selection : Roulette Wheel

GP의 초기 트리 생성에 대한 파라미터는 다음과 같다.

Initial population : half_and_half
 Initial depth : 1-3
 Max depth : 12

또한, GA는 각 모듈마다 3개씩 총 24개의 유전자를 하나의 개체로 생성하였으며, GP는 8개의 트리를 하나의 개체로 생성했다.

4.3 실험 결과 및 분석

각각의 생성된 함수에 대해 GA, GP 모두 한 사이클 당 40개의 보간점을 이용하였다. 각각 5회씩 반복 실험하였으며, 적합도가 클수록 단위시간당 이동거리가 긴 것을 의미한다.

실험은 뱀 로봇의 수직 이동 형태를 이용해서 이루어졌다. 적합도와 속도를 나타냈으며, 속도는 cm/s 로 측정하였다. 속도의 측정은 시뮬레이션의 변동성을 고려하고 결과의 안정성 비교를 위해 각 실험에서 얻어진 해에 대해서 5회씩 반복 수행시키면서 가장 높은 수치와 가장 낮은 수치를 빼 평균값을 기록하였다.

표 1에서 알수 있듯이 20 pop 과 30 pop 모두, GA가 GP보다 속도 면에서 빠른 것을 확인할 수 있다.

표 1. GA와 GP의 성능비교.

	GA		GP	
	적합도	속도	적합도	속도
20 pop	3.37	52.65	2.62	41.36
	2.92	45.62	1.52	23.81
	2.23	34.84	2.56	39.91
	2.83	44.21	2.84	44.42
	1.72	26.87	2.81	43.83
Avg	2.61	40.84	2.48	38.67
30 pop	3.44	53.75	2.12	33.15
	2.90	45.31	1.94	30.24
	3.68	57.50	3.45	53.87
	1.81	28.28	1.57	24.52
	3.00	46.97	3.51	54.96
Avg	2.97	46.36	2.52	39.35

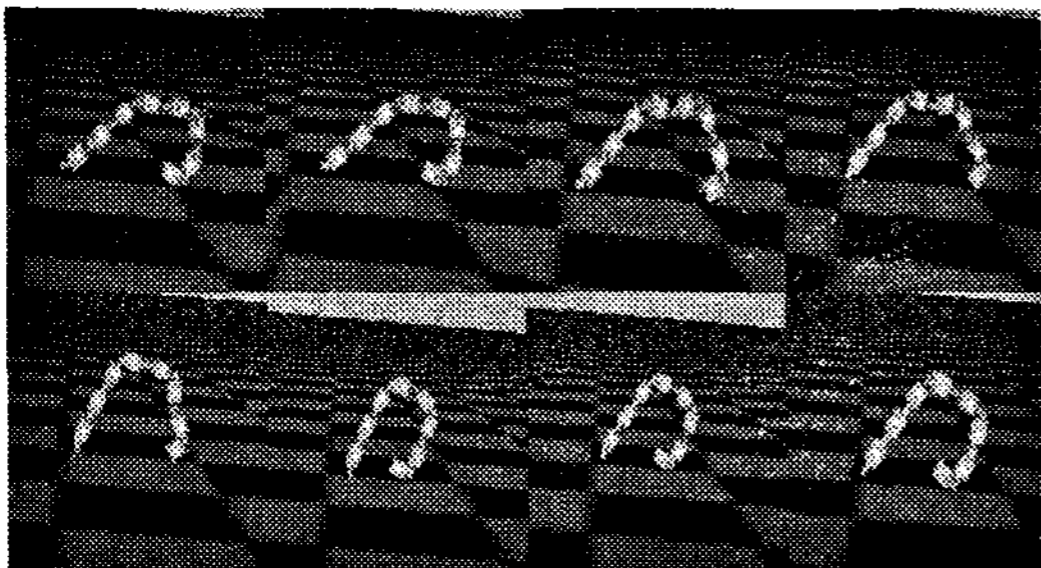


그림 5. GA(Pop=20) Best 개체의 움직임.

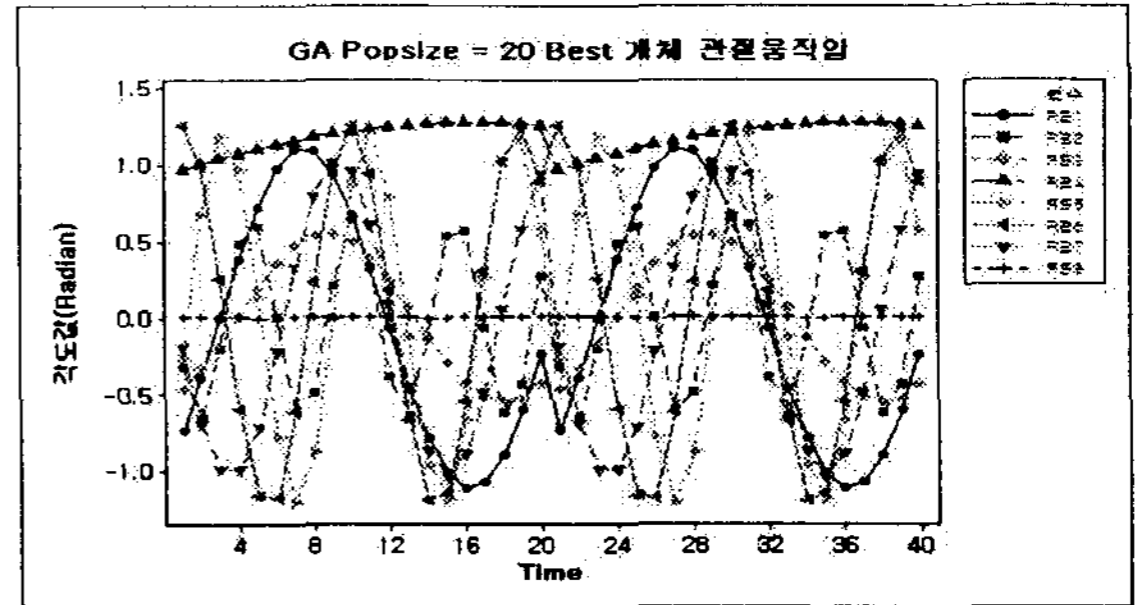


그림 6. GA(Pop=20) Best 개체 관절움직임.

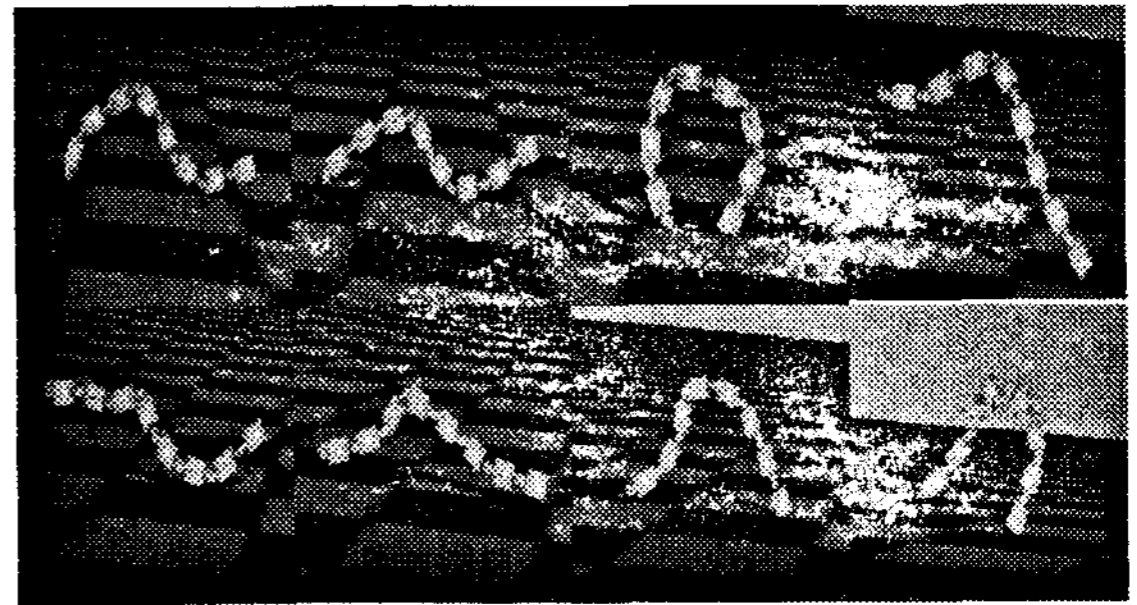


그림 7. GA(Pop=30) Best 개체의 움직임.

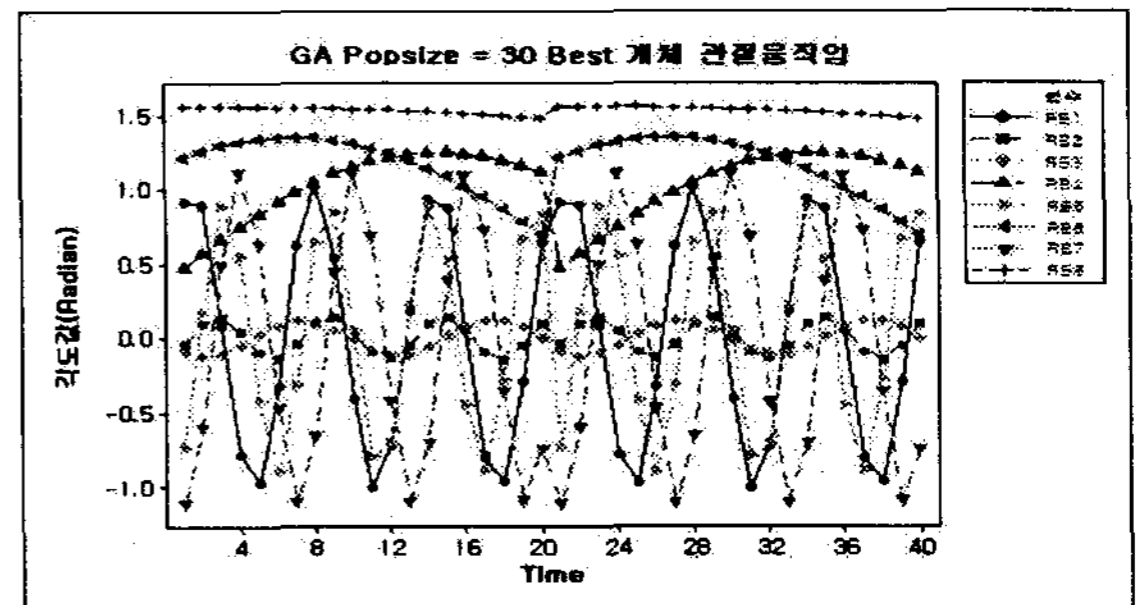


그림 8. GA(Pop=30) Best 개체의 관절움직임.

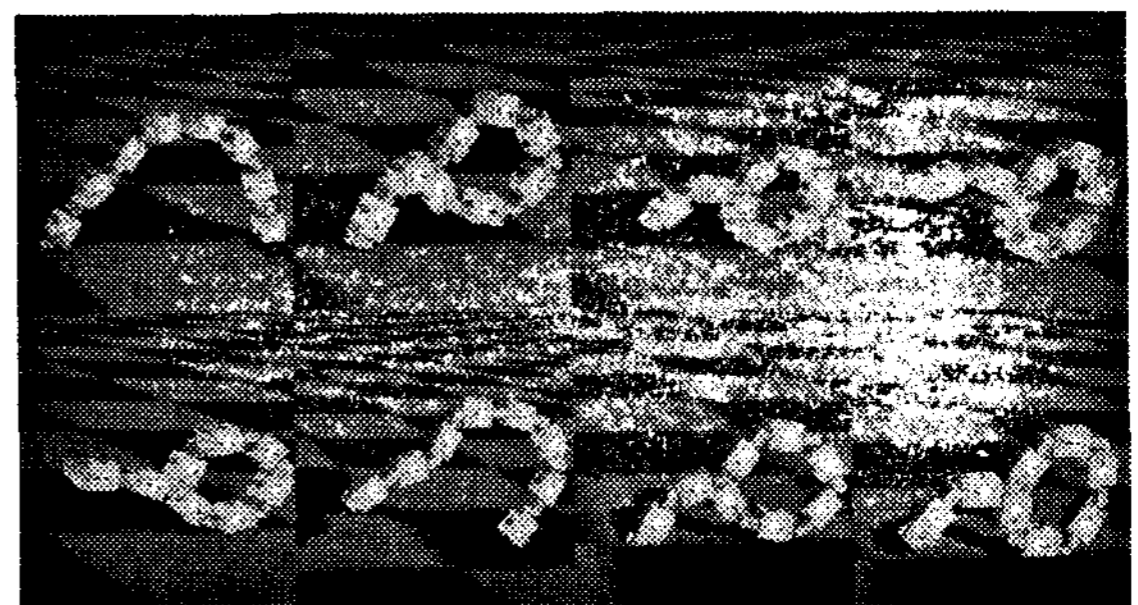


그림 9. GP(Pop=20) Best 개체의 움직임.

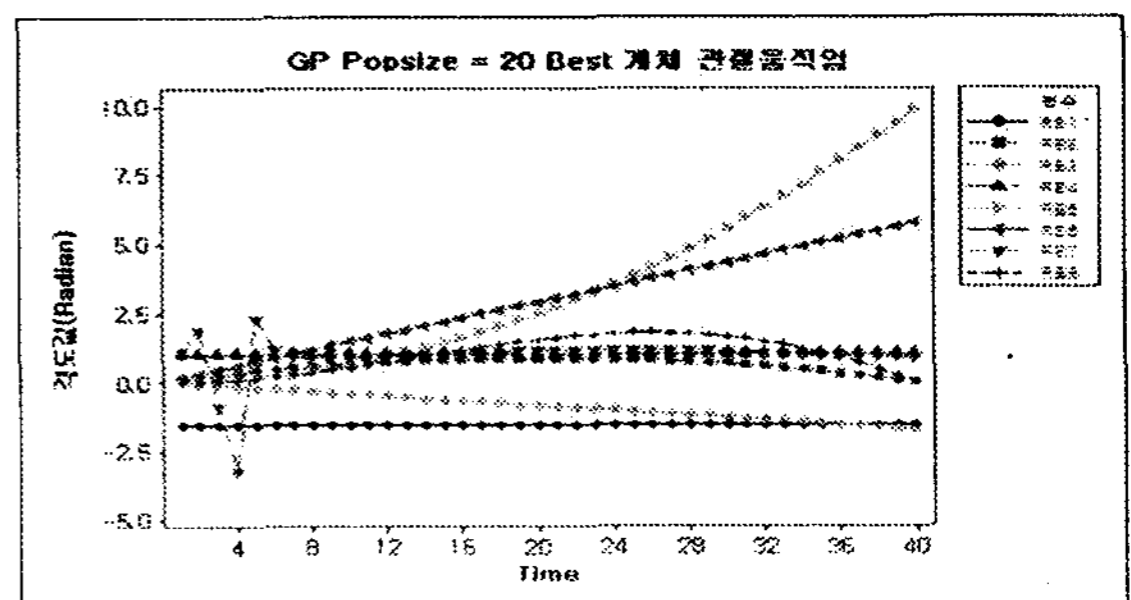


그림 10. GP(Pop=20) Best 개체의 관절움직임.

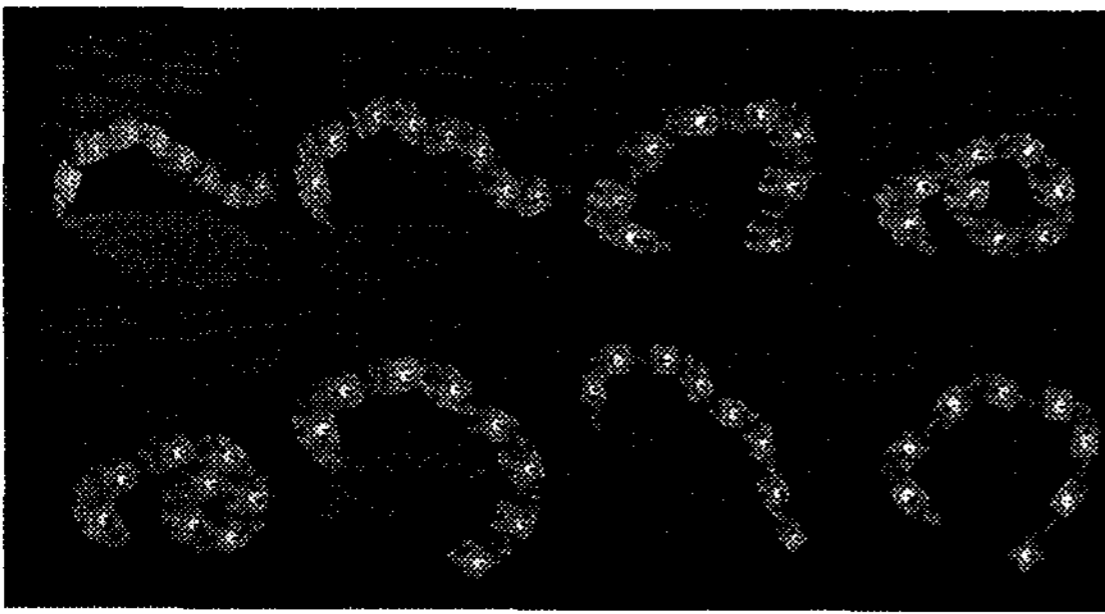


그림 11. GP(Pop=30) Best 개체의 움직임.

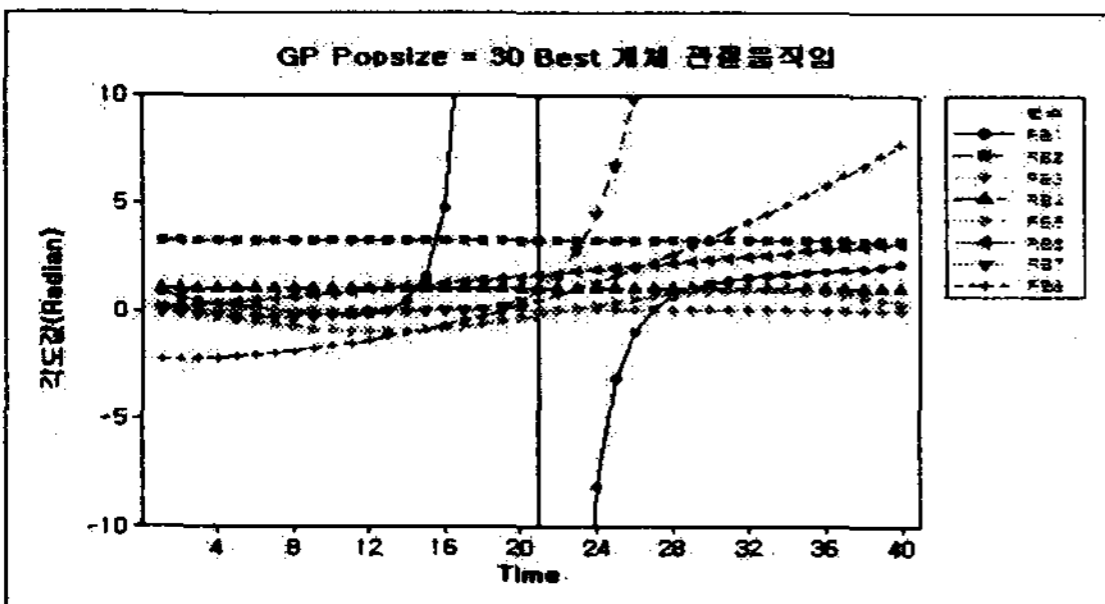


그림 12. GP(Pop=30) Best 개체 관절움직임.

GA와 GP의 최고적합도의 개체의 움직임을 비교해보면 (그림 5-12), GA, GP 양쪽 모두에서 몇몇 모듈의 순간적인 탄성을 이용한 형태의 움직임이 나타난 것을 확인할 수 있다. GA에서 몇몇 모듈은 아주 작은 움직임만을 가지기 때문에 그 영향이 미미함을 확인할 수 있다(그림 8). 또한, GP에서도 마찬가지로 몇몇 모듈은 고정적인 형태의 움직임을 갖는 것을 볼 수 있다(그림 10, 12).

하지만 GA에서는 주로 양 끝의 모듈의 탄성을 이용한 이동구조가 나타났으며(그림 5, 7), GP에서는 꼬리쪽의 4개의 모듈들을 이용한 이동 형태가 나타났다(그림 9, 11). GA에서는 그런 급격한 변화를 줄 수 없는 함수의 구조상 양 끝의 모듈을 활용한 움직임이 주로 나타나는 것으로 보인다. 또한, GP의 트리가 구조적으로 GA보다 더 큰 변동 값을 줄 수 있는 특성이 반영된 결과라고 보인다.

전반적으로, GP보다 GA에서 나타나는 움직임이 더 안정적이었다. GP에서는 어떤 개체들은 반복적인 실험 수행 시 동작이 불안정하게 되는 현상도 발생하였다.

아직 실험횟수와 GA/GP 파라미터에 변화에 대한 충분한 검토가 부족하기 때문에 두 가지 방법을 정확히 비교하기 어렵지만, GA의 경우 어느정도 관절 궤적이 일정한 패턴식을 변경한 것이므로, 상대적으로 안정적인 성능을 보인 것으로 판단된다. GP의 경우는 일반적으로 GA보다 많은 개체수를 필요로 하는 경향이 있고, 가변적인 트리의 특성상 구해지는 관절식에 대한 제한이 없으므로 훨씬 넓은 탐색 공간을 가진다. 따라서, 현재의 제한된 실험에서는 GA보다 약간 낮은 성능을 보이는 것으로 사료된다.

5. 결론

본 논문에서는 모듈라 뱀형 로봇에 GA와 GP를 이용한 이동제어를 통해 두 진화기법의 의한 성능 결과와 차이점에 대해서 분석하였다. 평균적으로 주요 성능지표인 속도 면에서 GA 쪽이 더 우수함을 나타내고, 이에 비해

GP의 경우는 움직임 면에서는 동작이 크고 정형화에서 벗어난 결과를 보인다. 개체수가 증가될 경우에는 GP가 속도면에서도 경쟁력 있는 결과를 보여줄 것으로 기대된다.

추후 연구로 수평이동 방식과, 수평과 수직을 동시에 움직이는 이동방식에 대한 확장과 함께 진화연산에 대한 개선 및 충분한 실험이 수반되어야 한다.

아울러, 모듈라 뱀형 로봇에 대한 시뮬레이션 결과를 이용하여, 실제로 구현된 모듈라 뱀형 로봇의 이동 제어에 적용하는 것이 필요하다.

참고 문헌

- [1] G. S. Hornby, H. Lipson, J. B. Pollack, "Generative Representations for the Automated Design of Modular Physical Robots", *IEEE Trans. on Robotics and Automation*, vol. 19, no. 4, pp. 703-719, 2003.
- [2] V. Zykov, E. Mytilinaios, M. Desnoyer, H. Lipson, "Evolved and Designed Self-Reproducing Modular Robotics", *IEEE Trans. on Robotics*, vol. 23, no. 2, pp. 308-319, 2007.
- [3] A. Crespi, A. J. Ijspeert, "AmphiBot II : An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator", In *Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006)*, pp. 19-27, Brussels, Belgium, September 2006.
- [4] I. Tanev, "Genetic programming incorporation biased mutation for evolution and adaptation of snakebot", *Genetic Programming and Evolvable Machines*, vol. 8, no. 1, pp. 39-59, 2007.
- [5] Mark W. Sherman, "Sine-Wave Locomotion in a Robotic Snake Model Form and Programming", In *Proceedings of AAI Mobile Robot Competition: Papers from the AAI Workshop*, pp. 34-40. American Association for Artificial Intelligence, Menlo Park, California, 2002.
- [6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [7] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT press, 1992.
- [8] L. Hohl, R. Tellez, O. Michel, A. J. Ijspeert, "Aibo and Webots: Simulation, wireless remote control and controller transfer", *Robotics and Autonomous Systems*, vol. 54, pp. 472-485, 2006.
- [9] A. Billard P. Dillenbourg, "Roombots." <http://birg.epfl.ch/page65721.html>.
- [10] K. J. Dowling, "Limbless Locomotion: Learning to Crawl with a Snake Robot", in his Ph.D Thesis, Robotics Institute, Carnegie Mellon University 1997.
- [11] H. Choset, "Modular Snake Robots" <http://download.srv.cs.cmu.edu/~biorobotics/projects/modsnake/modsnake.html>.
- [12] I. Tanev, T. Ray, K. Shimohara, "On the Analogy in the Emergent Properties of Evolved Locomotion Gaits of Simulated Snakebot", *Mobile Robots Toward New Applications*, chapter. 19, pp. 559-578, Germany, December 2006.
- [13] D. Zongker B. Punch, *Lil-GP User's Manual*. Michigan State University, July 1995.